while (!(cin >> golf[i])) {
    cin.clear(); // reset input
    while (cin.get() != '\n')
      continue; // get rid of bad input
    cout << "Please enter a number: ";
}

If the user enters 88, the cin expression is true, a value is placed in the array, the expression !(cin >> golf[i]) is false, and this inner loop terminates. But if the user enters must i?, the cin expression is false, nothing is placed into the array, the expression !(cin >> golf[i]) is true, and the program enters the inner while loop. The first statement in the loop uses the clear() method to reset input. If you omit this statement, the program refuses to read any more input. Next, the program uses cin.get() in a while loop to read the remaining input through the end of the line. This gets rid of the bad input along with anything else on the line. Another approach is to read to the next white space, which would get rid of bad input one word at a time instead of one line at a time. Finally, the program tells the user to enter a number.

**Summary**

Programs and programming become more interesting when you introduce statements that guide the program through alternative actions. (Whether this also makes the programmer more interesting is a point we've not fully researched.) C++ provides the if statement, the if else statement, and the switch statements as means for managing choices. The C++ if statement lets a program execute a statement or statement block conditionally. That is, the program executes the statement or block if a particular condition is met. The C++ if else statement lets a program select from two choices which statement or statement block to execute. You can append additional if elses to the statement to present a series of choices. The C++ switch statement directs the program to a particular case in a list of choices.

C++ also provides operators to help in decision making. Chapter 5 discusses the relational expressions, which compare two values. The if and if else statements typically use relational expressions as test conditions. By using C++'s logical operators (&&, ||, and !), you can combine or modify relational expressions, constructing more elaborate tests. The conditional operator (?:) provides a compact way to choose from two values.